

- **Functionality:** Applikationen skal opfylde kravene for CRUD-operationer, filtrering, sortering og præsentation af kunstnerdata. Funktionaliteten skal være pålidelig og intuitiv.

CRUD: Der er Create, Read, Update og Delete i min web-applikation og de er alle til at nemme at tilgå både front- og backend.

Filtrering: Det er nemt og hurtigt at navigere i mit sortering- samt filtreringsværktøj som man kan finde øverst på websiden. Hvis man klikker på "Favorite"-knappen kan man filtrere gennem ens favorite-kunstnere.

Sortering: Man kan sortere nemt ved at klikke på "Sort By", hvori der kommer en drop-down menu med de valgmuligheder der er sat op. Disse er også efter alfabetisk rækkefølge.

Præsentation af kunstner data: Disse kan man få vist som navne i en liste hvis man klikker på den pågældende kunstner. Der åbnes et modal vindue op hvor kunstnerens informationer står.

- **Usability:** Brugergrænsefladen skal være brugervenlig og nem at navigere. Alle CRUD-operationer skal være let tilgængelige, og filtrerings/sorteringsfunktionerne skal være intuitive

Min brugergrænseflade er nem at navigere i, da der er overblik lige fra starten når man kører appen. Sortering-, filtrering- og favoritfunktionerne er alle nemme at bruge og forstå, da disse valgmuligheder står på samme sted i min applikation.

- **Reliability:** Applikationen skal være stabil og pålidelig. Dataintegritet og korrekt håndtering af CRUD-operationer er afgørende.

Appen giver fejlbeskeder tilbage til modtageren når den skal gøre det og min API er specifik i hvad den godtager i dens operationer.

- **Performance:** Applikationen skal have en acceptabel ydeevne, herunder hurtig datahentning og responsivitet i brugergrænsefladen.

Webapplikationen er ret hurtig når man sletter, læser, klikker, opretter og opdaterer kunstnerne. Dens API kald fungerer som de skal.

- **Supportability:** Koden skal være velstruktureret og veldokumenteret, så det er nemt for fremtidige udviklere at vedligeholde og udvide applikationen.

Der er opsat kommentarer op i koden for at gøre det nemt for udviklerne at rette i min kode, og funktionerne har navne, som er med til at hjælpe den som læser min kode med at forstå hvad koden gør.

- **+:** Der skal også tages hensyn til andre relevante krav og overvejelser for at sikre en vellykket udvikling og implementering af applikationen. Ud over FURPS-kravene skal

udviklingen også overholde generelle principper for kodestruktur, modularitet og afhængighedsstyring, herunder Separation of Concerns, Loose Coupling og High Cohesion.

Funktionerne er bygget op på den måde at de fleste er bundet sammen med et API. Det betyder at funktionerne skulle ændres hvis der bliver tilgået et API kald. Funktionerne er så uafhængige af hinanden som muligt.